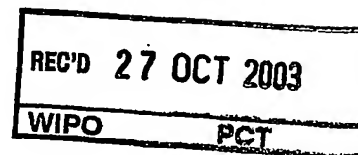**PRIORITY DOCUMENT**

SUBMITTED OR TRANSMITTED IN
COMPLIANCE WITH RULE 17.1(a) OR (b)

**Patent Office
Canberra**

I, JULIE BILLINGSLEY, TEAM LEADER EXAMINATION SUPPORT AND
SALES hereby certify that annexed is a true copy of the Provisional specification
in connection with Application No. 2002952259 for a patent by
SILVERBROOK RESEARCH PTY. LTD. as filed on 25 October 2002.

WITNESS my hand this
Twenty-third day of October 2003

J. Billingsley

JULIE BILLINGSLEY
TEAM LEADER EXAMINATION
SUPPORT AND SALES

Best Available Copy

AUSTRALIA

Patents Act 1990

Provisional Specification

for an invention entitled:

METHODS AND APPARATUS (NPT019)

The invention is described in the following statement: -

METHODS AND APPARATUS (NPT019)

1 Introduction

This document defines the surface coding used by the netpage system (as disclosed in the present applicants' co-pending PCT application publication number WO 01/22207 - Method and System for Instruction of a Computer, the contents of which is herein incorporated by reference) to imbue otherwise passive surfaces with interactivity in conjunction with netpage sensing devices such as the netpage pen (as disclosed in the present applicants' co-pending PCT application publication number WO 00/72230 - Sensing Device, the contents of which is herein incorporated by reference) and the netpage viewer (as disclosed in the present applicants' co-pending PCT application publication number WO 01/41046 - Viewer with Code Sensor, the contents of which is herein incorporated by reference).

When interacting with a netpage coded surface, a netpage sensing device generates a digital ink stream which indicates both the identity of the surface region relative to which the sensing device is moving, and the absolute path of the sensing device within the region.

2   Surface Coding

The netpage surface coding consists of a dense planar tiling of tags. Each tag encodes its own location in the plane. Each tag also encodes, in conjunction with adjacent tags, an identifier of the region containing the tag. In the netpage system, the region typically corresponds to the entire extent of the tagged surface, such as one side of a sheet of paper.

Each tag is represented by a pattern which contains two kinds of elements. The first kind of element is a target. Targets allow a tag to be located in an image of a coded surface, and allow the perspective distortion of the tag to be inferred. The second kind of element is a macrodot. Each macrodot encodes the value of a bit by its presence or absence.

The pattern is represented on the coded surface in such a way as to allow it to be acquired by an optical imaging system, and in particular by an optical system with a narrowband response in the near-infrared. The pattern is typically printed onto the surface using a narrowband near-infrared ink.

2.1   TAG STRUCTURE

Figure 1 shows the structure of a complete tag. Each of the six black circles is a target. The tag, and the overall pattern, is six-fold symmetric at the physical level.
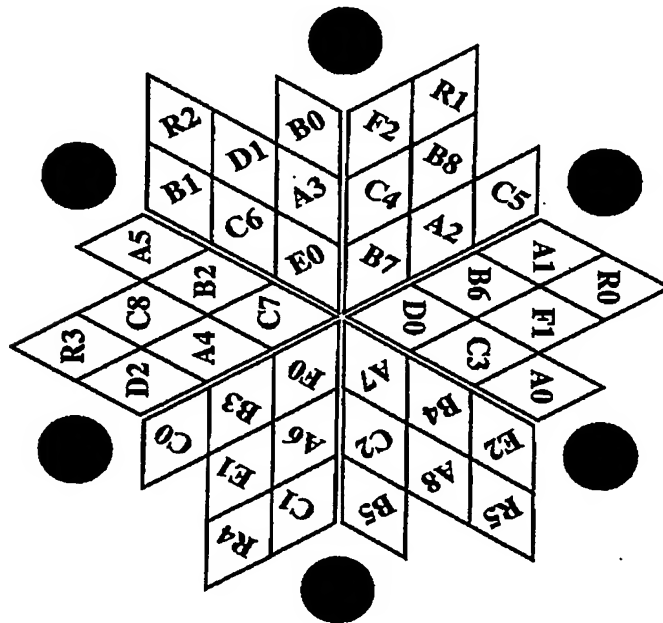

Figure 1. Tag structure

Each diamond-shaped region represents a symbol, and each symbol represents four bits of information.

Figure 2 shows the structure of a symbol. It contains four macrodots, each of which represents the value of one bit by its presence (one) or absence (zero).

The macrodot spacing is specified by the parameter s throughout this document. It has a nominal value of $143\mu m$, based on 9 dots printed at a pitch of 1600 dots per inch. However, it is allowed to vary within defined bounds according to the capabilities of the device used to produce the pattern.

In general, if a surface is coded with a pattern which deviates from the "ideal" pattern specified in this document, e.g. due to device limitations, then the deviation must be recorded so that any digital ink captured via the surface can be appropriately corrected for the deviation.
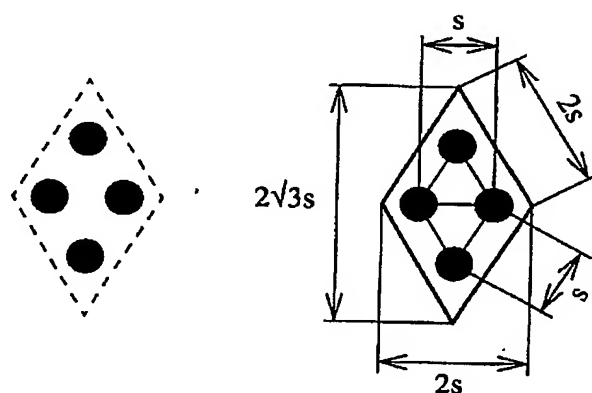

Figure 2. Symbol unit cell

Figure 3 shows an array of five adjacent symbols. The macrodot spacing is uniform both within and between symbols.
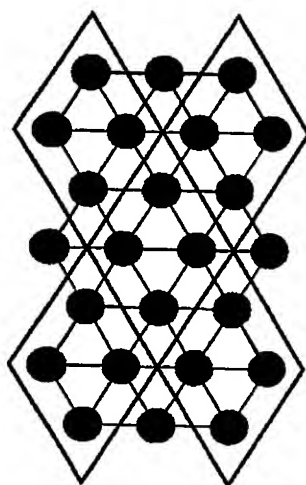

Figure 3. Array of symbol unit cells

Figure 4 shows the ordering of the bits within a symbol. Bit zero is the least significant within a symbol; bit three is the most significant. Note that this ordering is relative to the orientation of the symbol. The orientation of a particular symbol within the tag is indicated by the orientation of the label of the symbol in the tag diagrams. In general, the orientation of all symbols within a particular segment of the tag have the same orientation, consistent with the bottom of the symbol being closest to the centre of the tag.
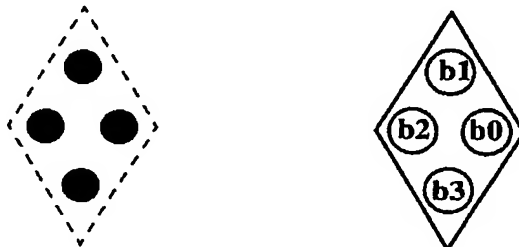

Figure 4. Symbol bit ordering

Only the macrodots are part of the representation of a symbol in the pattern. The diamond-shaped outline of a symbol is used in this document to more clearly elucidate the structure of a tag. Figure 5, by way of illustration, shows the actual pattern of a tag with every bit set. Note that, in practice, every bit of a tag can never be set.
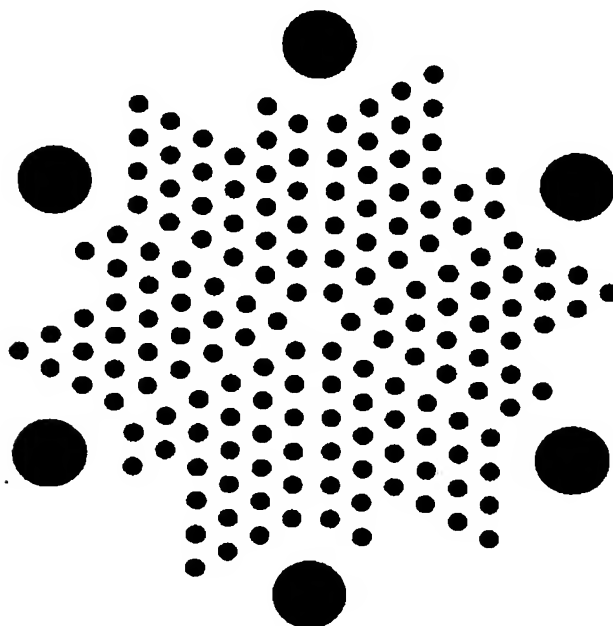

Figure 5. Tag structure with every bit set

A macrodot is nominally circular with a nominal diameter of $(5/9)s$. However, it is allowed to vary within defined bounds according to the capabilities of the device used to produce the pattern.

A target is nominally circular with a nominal diameter of $(17/9)s$. However, it is allowed to vary within defined bounds according to the capabilities of the device used to produce the pattern.

Each symbol shown in the tag structure in Figure 1 has a unique label. Each label consists an alphabetic prefix and a numeric suffix.

2.2 ORIENTATION-INDICATING CYCLIC POSITION CODE

The tag contains a ternary $(6, 1)$ cyclic position codeword (as disclosed in the present Applicants' co-pending PCT application publication number WO 02/084473 - Cyclic Position Codes, the contents of which is herein incorporated by reference) which can be decoded at any of the six possible orientations of the tag to determine the actual orientation of the tag. Symbols which are part of the cyclic position codeword have a prefix of "R" and are numbered 0 to 5 in order of increasing significance.

The cyclic position codeword is $(1, 2, 3, 4, 5, 6)$. Note that it only uses six distinct symbol values, even though a four-bit symbol has sixteen possible values. During decoding, any unused symbol value should, if detected, be treated as an erasure.

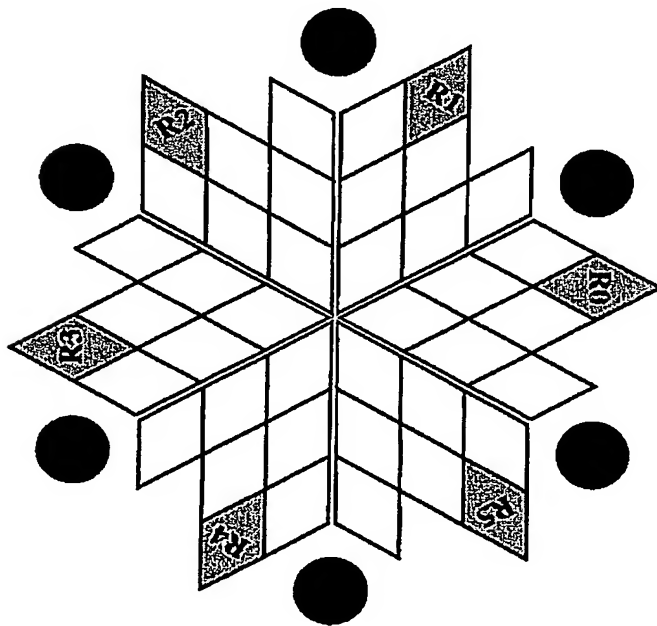The layout of the orientation-indicating cyclic position codeword is shown in Figure 6.

Figure 6. Orientation-indicating cyclic position codeword R (codeword is shown shaded)

2.3   LOCAL CODEWORDS

The tag locally contains three complete codewords which are used to encode information unique to the tag. Each codeword is of a punctured quaternary (9, 5) Reed-Solomon code. The tag therefore encodes up to 60 bits of information unique to the tag.

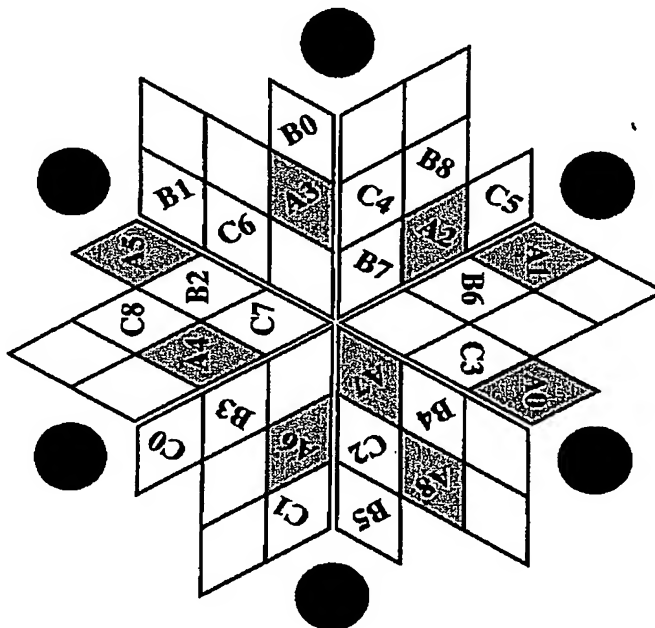The layout of the three local codewords is shown in Figure 7.

**Figure 7. Interleaved local codewords A, B and C
(codeword A is shown shaded)**

2.4   DISTRIBUTED CODEWORDS

The tag also contains fragments of three codewords which are distributed across three adjacent tags and which are used to encode information common to a set of contiguous tags. Each codeword is of a punctured quaternary (9, 5) Reed-Solomon code. Any three adjacent tags therefore together encode up to 60 bits of information common to a set of contiguous tags.

The layout of the three codeword fragments is shown in Figure 8.

The layout of the three complete codewords, distributed across three adjacent tags, is shown in Figure 9. In relation to these distributed codewords there are three types of tag. These are referred to as P, Q and R in order of increasing significance.

Figure 10 shows how the P, Q and R tags are repeated in a continuous tiling of tags. The tiling guarantees the any set of three adjacent tags contains one tag of each type, and therefore contains a complete set of distributed codewords. The tag type, used to determine the registration of the distributed codewords with respect to a particular set of adjacent tags, is encoded in one of the local codewords of each tag.
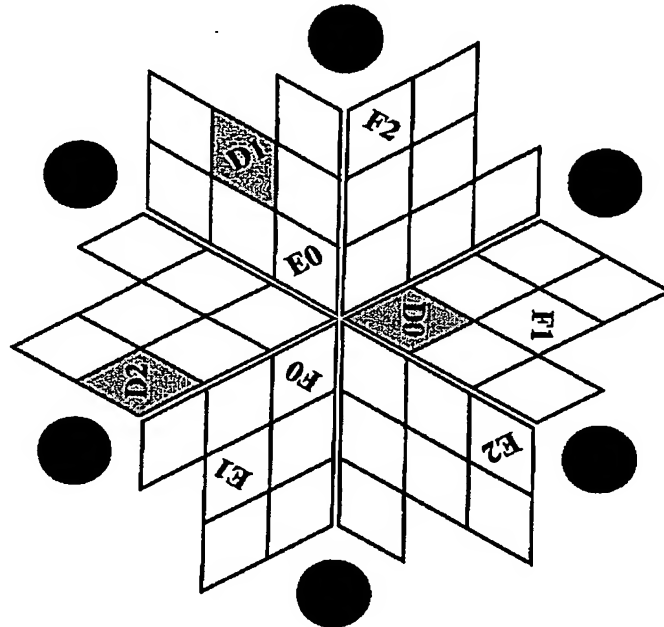
Figure 8. Interleaved fragments of distributed codewords D, E and F
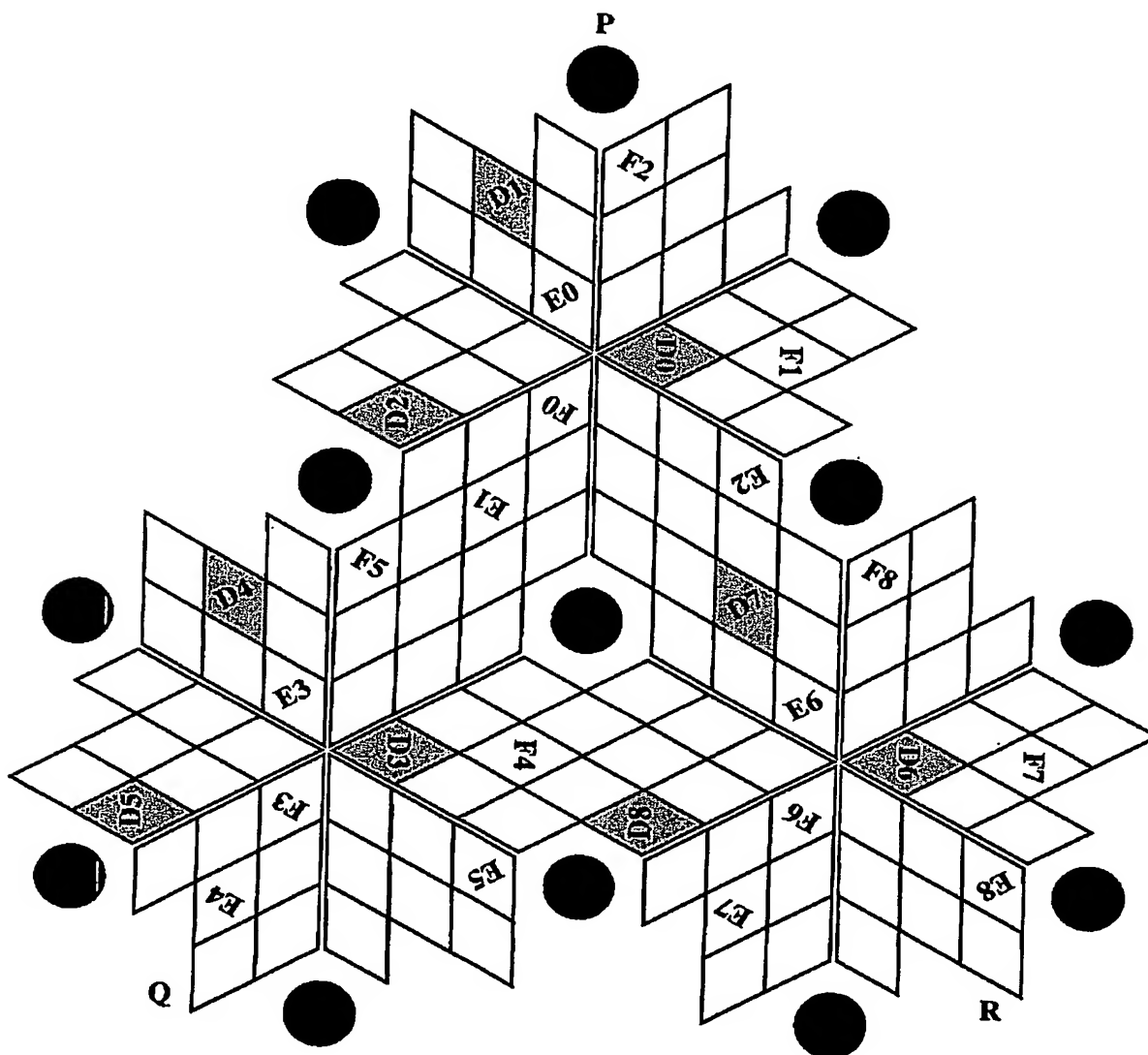(fragment of codeword D is shown shaded)

Figure 9. Three adjacent tags P, Q and R
containing a complete set of distributed codewords D, E and F
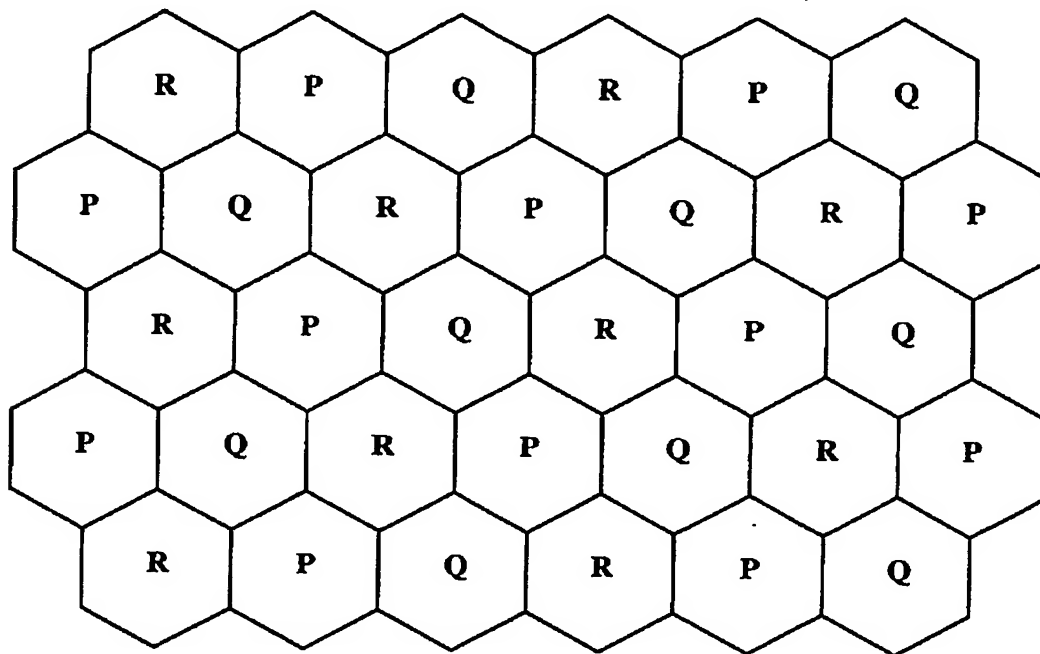(codeword D is shown shaded)

Figure 10. Continuous tiling of tags P, Q and R

Figure 11 shows the complete structure of three adjacent tags, including their orientation cyclic position codewords, local codewords and distributed codewords.
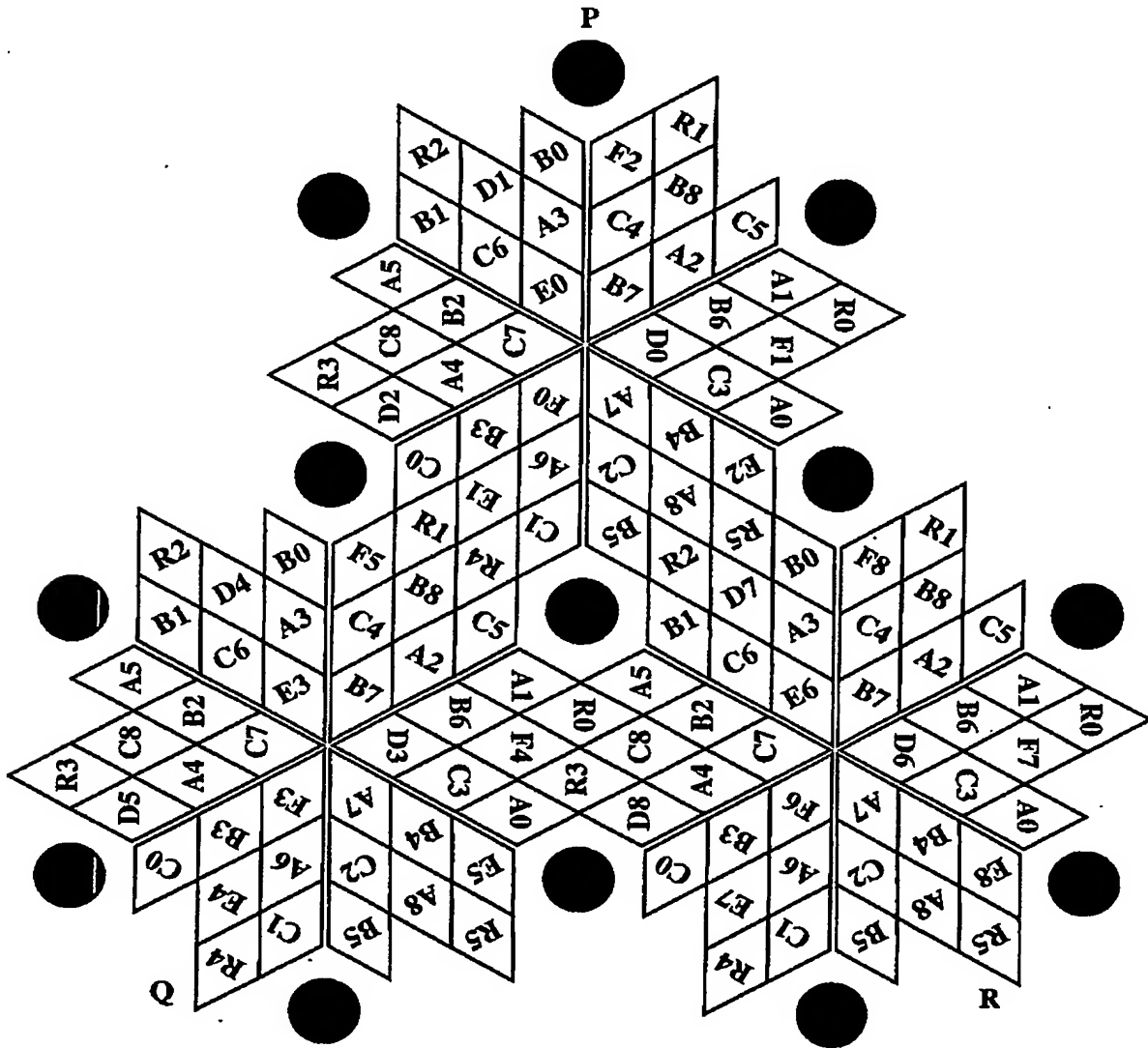
**Figure 11. Three adjacent tags P, Q and R
containing three complete sets of local codewords A, B and C,
and a complete set of distributed codewords D, E and F**

Figure 12 shows the geometry of a tag segment.

Figure 13 shows the spacing $d = (1 - \sqrt{3}/2)s$ between tag segments, required to maintain consistent spacing between macrodots.

Figure 14 shows the effect of the inter-segment spacing d on target position. Compared with their nominal positions in relation to closely-packed segments (i.e. with $d = 0$), diagonal targets must be displaced by $(\Delta_x, \Delta_y) = (\pm 1/\sqrt{3}, \pm 1)d$, and horizontal targets must be displaced by $(\Delta_x, \Delta_y) = (\pm 2/\sqrt{3}, 0)d$.
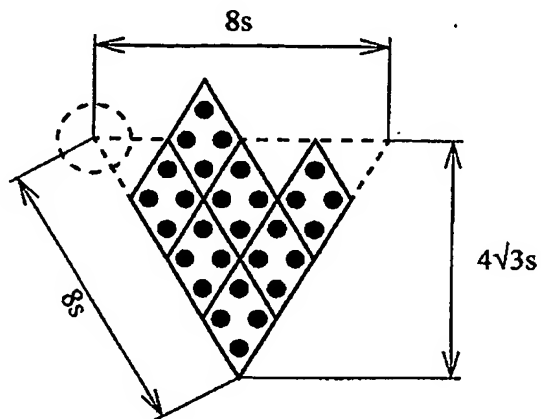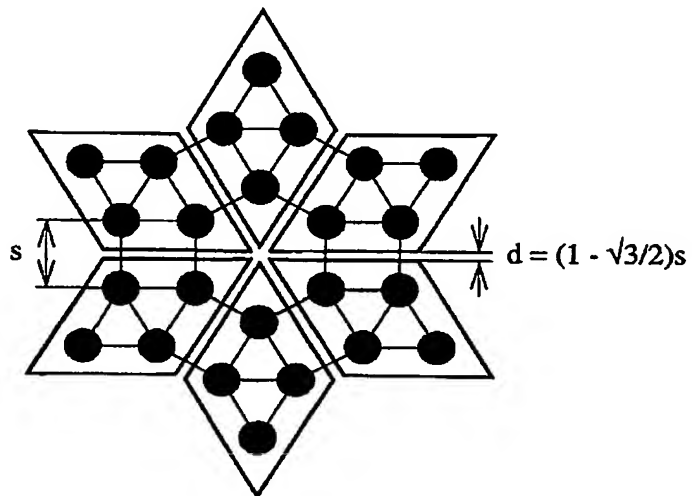
Figure 12. Tag segment

**Figure 13. Inter-segment spacing
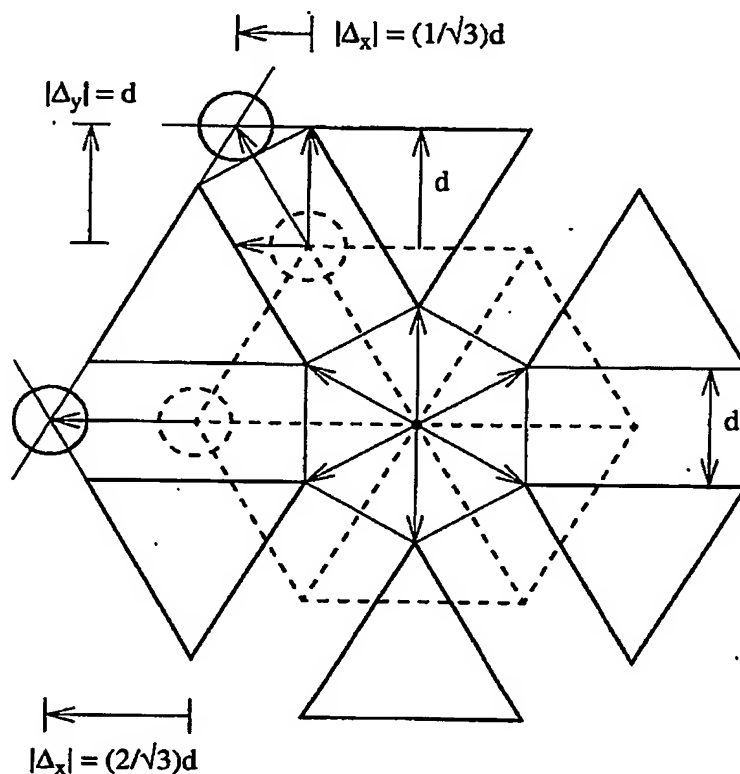(showing inner-most symbols of each segment only)**

Figure 14. Effect of inter-segment spacing on target position (with exaggerated inter-segment spacing for clarity)

2.5   REED-SOLOMON ENCODING

Both local and distributed codewords are encoded using a punctured quaternary $(9, 5)$ Reed-Solomon code.

A quaternary $(9, 5)$ Reed-Solomon code encodes 20 data bits (i.e. five 4-bit symbols) and 16 redundancy bits (i.e. four 4-bit symbols) in each codeword. Its error-detecting capacity is four symbols. Its error-correcting capacity is two symbols.

A punctured quaternary $(9, 5)$ Reed-Solomon code is a quaternary $(15, 5)$ Reed-Solomon code with six redundancy coordinates removed.

The code has the following generator polynominal:

$$g(x) = (x + \alpha)(x + \alpha^2)...(x + \alpha^{10})$$ (EQ 1)

For a general discussion of Reed-Solomon codes, refer to Wicker, S.B. and V.K. Bhargava, eds., *Reed-Solomon Codes and Their Applications*, IEEE Press, 1994. For examples of Reed-Solomon error and erasure decoders, refer to Morelos-Zaragoza, R., *The Error Correcting Codes (ECC) Page*, http://www.eccpage.com.

3 Tag Coordinate Space

The tag coordinate space is defined by a pair of semi-orthogonal coordinates a and b. The nominal relationship between the tag coordinate space and the surface x-y coordinate space is illustrated in Figure 15.
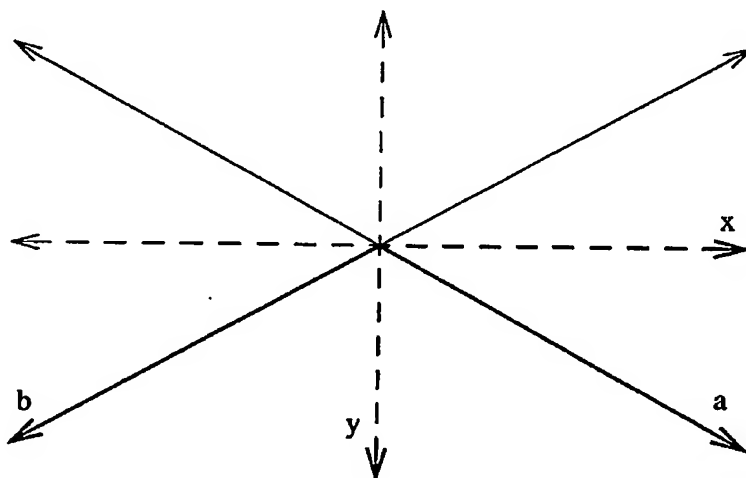

Figure 15. Relationship of tag a-b coordinate space to surface x-y coordinate space

The coordinates are necessarily only semi-orthogonal to ensure reproducibility of the tag pattern by intended printing devices (as discussed in Appendix A).

To further assist intended printing devices, the surface coding, and hence the a-b coordinate space, is allowed to be rotated an arbitrary multiple of 90 degrees with respect to the x-y coordinate space. The actual rotation of a surface coding is recorded in the coding.

Given an anti-clockwise rotation R of the a-b coordinate space with respect to the x-y coordinate space, the relations between the two coordinates spaces are:

$$x = a\cos(30° - R) - b\sin(60° - R) \qquad \text{(EQ 2)}$$

$$y = a\sin(30° - R) + b\cos(60° - R) \qquad \text{(EQ 3)}$$

$$a = \frac{x}{2\cos(30° - R)} + \frac{y}{2\sin(30° - R)} \qquad \text{(EQ 4)}$$

$$b = \frac{-x}{2\sin(60° - R)} + \frac{y}{2\cos(60° - R)} \qquad \text{(EQ 5)}$$

Integer a and b coordinates are defined to intersect at the centres of P tags, as illustrated in Figure 16.

Note that the surface coding does not specify the location of the x-y (or a-b) origin on a particular tagged surface, or the orientation of the x-y (or a-b) coordinate space with respect to the surface. It only defines the relationship between the two coordinate spaces.

This manifests itself in the x-y coordinates embedded in digital ink generated by a netpage sensing device used to interact with a netpage tagged surface.
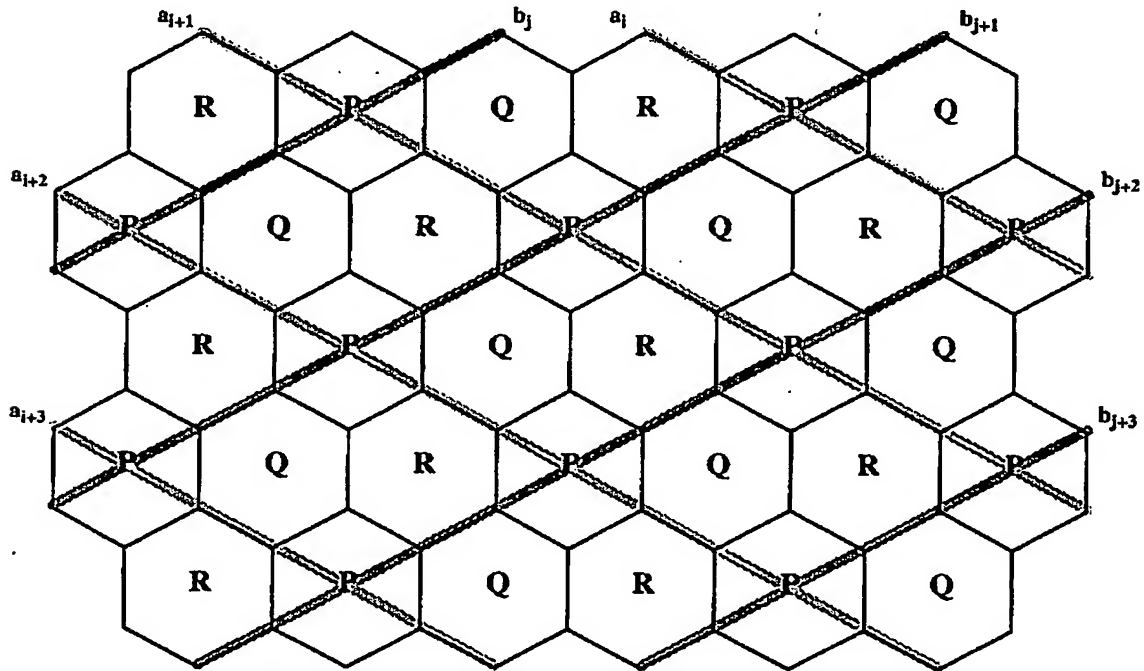

**Figure 16. Tag coordinate grid superimposed on tag tiling**

4   Tag Information Content

Table 1 defines the information fields embedded in the surface coding. Table 2 defines how these fields map to local and distributed codewords.

Table 1. Field definitions

field	width	description
tag type	2	Defines whether the tag is of type P (b'00'), Q (b'01') or R (b'10').
tag rotation	2	The anti-clockwise rotation of the tag with respect to the x-y coordinate space; in units of 90 degrees.
local flag	1	Indicates whether the region ID is owned by a local netpage system (b'1') or the global netpage system (b'0').
reserved	1	Reserved; must be zero.
active area map	7	A map[a] of which of the tag and its immediate neighbours are members of an active area; b'1' indicates membership.
input area map	7	A map[a] of which of the tag and its immediate neighbours are members of an input area; b'1' indicates membership.
coordinate precision (w)	5	The precision of the a and b coordinates; valid range is 0 to 20.
a coordinate	w (0 to 20)	The signed a coordinate of the tag, in sign-magnitude format.
b coordinate	w (0 to 20)	The signed b coordinate of the tag, in sign-magnitude format.
region ID	95–2w (95 to 55)	The ID of the region containing the tags.
total	120	

a. Figure 17 indicates the bit ordering of the map.
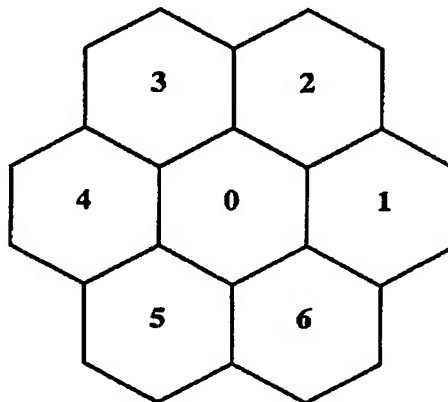
Figure 17. A tag and its six immediate neighbours,
each labelled with its corresponding bit index in the active area map and the input area map

Since the top 55 bits of the region ID are encoded in distributed codewords, they are by necessity constant for an entire contiguous tiling of tags. The bottom 40 bits, however, are encoded in local codewords, and so may vary arbitrarily from one tag to the next.

For a particular surface coding, the number of bits dedicated to the a and b coordinates is configurable via the coordinate precision field. In this way the precision can be tuned to the size of the surface being tagged, which in turn allows efficient use of a higher-precision region ID space. Region IDs can be allocated from the full-precision 95-bit space, but with the constraint that for a particular coordinate precision of w, the bottom $2w$ bits of each allocated region ID must be zero. Almost equivalently, different-precision region IDs can be allocated from precision-specific pools, and each allocated region ID can be indexed and looked-up in a pool-specific way.

Table 2. Mapping of fields to codewords

field	codeword	codeword width	codeword bits	field bits
tag type	A	2	1:0	all
tag rotation		2	3:2	all
local flag		1	4	all
reserved		1	5	all
active area map		7	12:6	all
input area map		7	19:13	all
coordinate precision (w)	D	5	19:15	all
a coordinate	B	w	$(w-1):0$	all
b coordinate	C	w	$(w-1):0$	all
region ID	D	15	14:0	94:80
	E	20	19:0	79:60
	F	20	19:0	59:40
	B	20-w	$19:w$[a]	$39:(20+w)$[a]
	C	20-w	$19:w$[a]	$(19+w):2w$[a]

a. If $w < 20$

The active area map indicates whether the corresponding tags are members of an active area. An active area is an area within which any captured input should be immediately forwarded to the corresponding netpage server for interpretation. It also allows the netpage sensing device to signal to the user that the input will have an immediate effect.

The input area map indicates whether the corresponding tags are members of an input area, i.e. lie within the extent of a form. It allows the netpage sensing device to signal to the user that the input will be submitted to an application.

Appendix A: Encoding and Printing Considerations

The Print Engine Controller (PEC) (as disclosed in the present Applicants' co-pending PCT application publication numbers WO 01/89851 - Print Engine/Controller and Print-head Interface Chip Incoroporating the Print Engine/Controller and WO 01/89838 - Printed Page Tag Encoder, the contents of both of which is herein incorporated by reference) supports the encoding of two fixed (per-page) quaternary $(15, 5)$ Reed-Solomon codewords and six variable (per-tag) quaternary $(15, 5)$ Reed-Solomon codewords. Furthermore, PEC supports the rendering of tags via a rectangular unit cell whose layout is constant (per page) but whose variable codeword data may vary from one unit cell to the next. PEC does not allow unit cells to overlap in the direction of page movement.

Figure 18 shows a proposed PEC unit cell compatible with the present surface coding, superimposed on a tiling of tags. Figure 19 shows the proposed PEC unit cell superimposed on actual P, Q and R type tag structures. Note that the proposed unit cell is centered horizontally on a P type tag. The tag structure and unit cell are designed so that the unit cell contains exactly six variable codewords (i.e., for row n, $R_n{:}A$, $P_n{:}A$, $Q_n{:}A$, $\{R,P,Q\}_n{:}B$, $\{R,P,Q\}_n{:}C$, and $\{R,P,Q\}_{n+1}{:}B$), and three fixed codewords D, E and F.
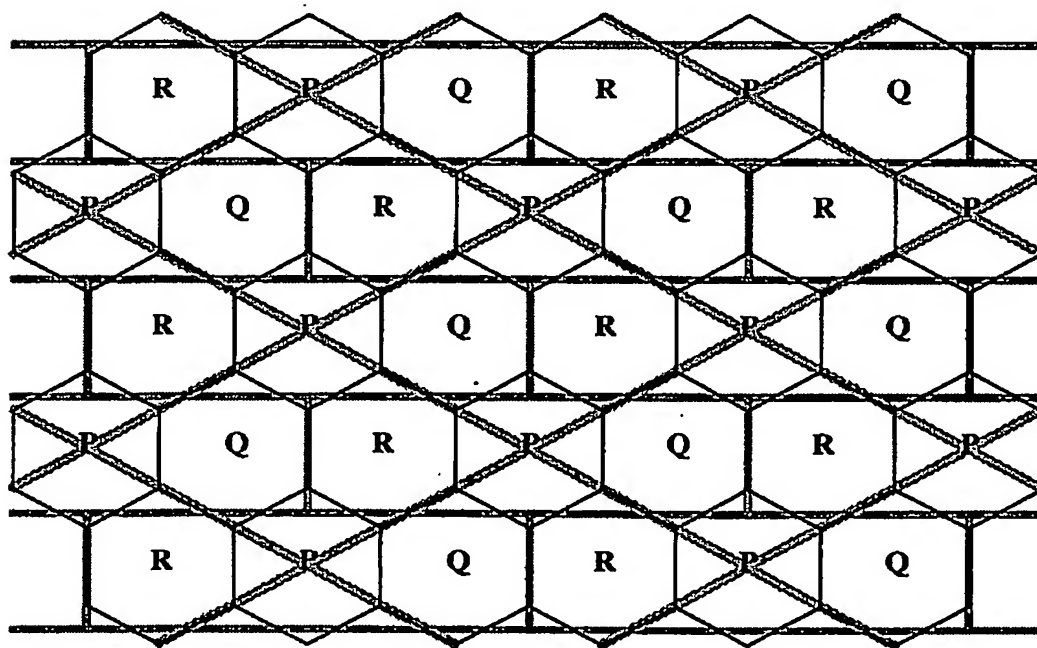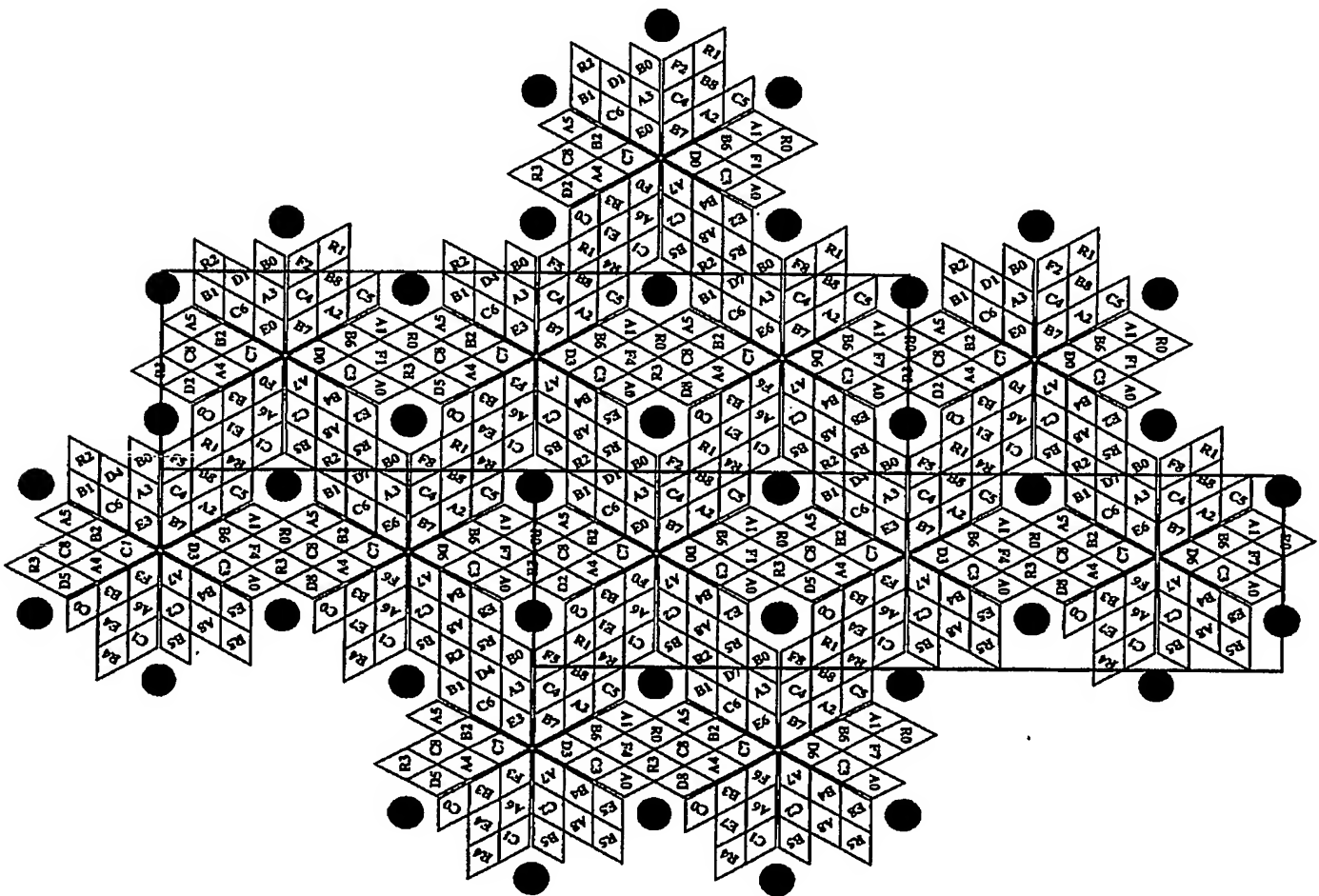

Figure 18. PEC unit cells (in red) superimposed on tag tiling, with tag coordinate grid in background (in gray)

At least one of codewords D, E and F must be pre-encoded in the Tag Format Structure (TFS) passed to PEC, since PEC only supports the encoding of two fixed codewords. Any or all of codewords D, E and F could be pre-encoded in the TFS.

PEC imposes a limit of 32 unique bit addresses per TFS row. The contents of the unit cell respect this limit, assuming pre-encoding of D, E and F codewords.

PEC also imposes a limit of 384.on the width of the TFS. The contents of the unit cell respect this limit.

Note that for a reasonable page size, the number of variable coordinate bits in the B and C codewords is modest, making encoding via a lookup table tractable. Encoding of the A codeword via a lookup table may also be possible. Note that since a Reed-Solomon code is systematic, only the redundancy data needs to appear in the lookup table.

Figure 19. PEC unit cells superimposed on tag structures

Appendix B: Imaging and Decoding Considerations

Figure 20 shows the minimal imaging field of view required to guarantee acquisition of an entire tag, i.e. given arbitrary alignment between the surface coding and the field of view.

The diameter of the minimal field of view is $36s$. Given worst-case tilt of $30°$ and three-times oversampling, this implies an image sensor size of 125 pixels.

Given the present tag structure, the corresponding decoding sequence is as follows:

- locate targets of complete tag
- infer perspective transform from targets
- sample cyclic position code
- decode cyclic position code
- determine orientation from cyclic position code
- sample local Reed-Solomon codewords
- decode local Reed-Solomon codewords
- determine tag type
- determine tag rotation
- sample distributed Reed-Solomon codewords
  (modulo window alignment, with reference to tag type)
- decode distributed Reed-Solomon codewords
- determine coordinate precision
- determine region ID
- determine tag a-b location
- transform tag a-b location to x-y location, with reference to tag rotation
- infer 3D transform from oriented targets
- determine nib x-y location from tag x-y location and 3D transform
- determine active/input area status of nib location
- generate local feedback based on nib active/input area status
- encode region ID, nib x-y location, nib active/input area status in digital ink
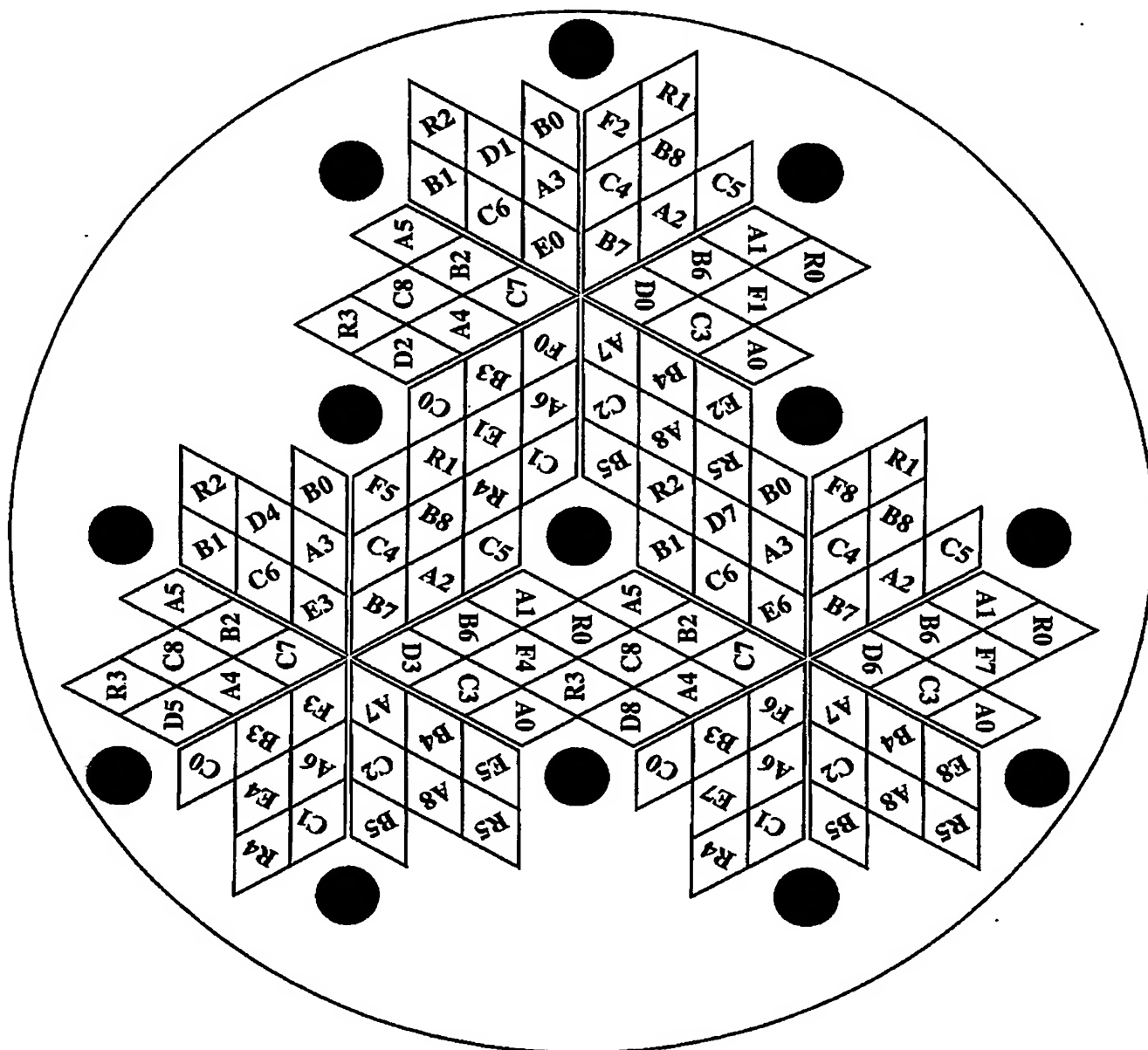
Figure 20. Minimal field of view

CLAIMS

1.   A method of determining a rotational orientation of a coded data structure disposed on or in a substrate, the coded data structure including at least a first codeword of a first cyclic position code, the cyclic position code having n symbols and a minimum distance d_{min}, the n symbols of the first codeword being disposed in s groups of n/s symbols, the s groups being spaced equi-angularly about a centre of the coded data structure, the method including the steps of:
        sensing the data structure to obtain the first codeword;
        obtaining from the first codeword a first subsequence of length w symbols, where $w \geq n - d_{min} + 1$;
        mapping the first subsequence to a second codeword of the first cyclic position code; and
        determining an offset of the second codeword with respect to the first codeword; and
        determining, from the offset, the relative rotational orientation of the coded data structure.

2.   A method of decoding data disposed within a coded data structure disposed on or in a substrate, the method including the steps of:
        determining a relative rotational orientation of the coded data structure in accordance with the method of claim 1; and
        decoding the data with reference to the relative rotational orientation.

3.   A method according to claim 2, wherein the data and the first codeword are sensed simultaneously.

4.   A method according to claim 2 or 3, wherein there are at least two pieces of data, the two pieces of data being disposed equi-angularly about the axis of the coded data structure in respective identical layout patterns.

5.   A method according to any one of the preceding claims, wherein $n = s$.

6.   A method according to any one of the preceding claims, wherein $n = 2, 3, 4$ or 6.

7.   A substrate bearing one or more coded data structures according to any one of the preceding claims.

8.   A substrate according to claim 7, wherein the coded data structures are printed onto a surface of the substrate.

9.   A substrate according to claim 7 or 8, wherein the coded data structures are printed in an infra-red absorbent ink that is invisible or substantially invisible to an average unaided human.

10.   A method of manufacturing a substrate according to any one of claims 7 to 9, including printing one or more of the coded data structures onto a surface of the substrate.

11.   A coded data structure for disposition on or in a substrate, the coded data structure including at least a first codeword of a first cyclic position code, the cyclic position code having n symbols and a minimum distance d_{min}, the n symbols of the first codeword being disposed in s groups of n/s symbols, the s groups being spaced equi-angularly about an axis of the coded data structure, such that a rotational orientation of the coded data structure can be determined by the method of any of the preceding claims

12.   A plurality of coded data structures, each of the coded data structures including at least one local codeword and some portion of a distributed codeword, the plurality of coded data structures being disposed on or in a substrate in a arrangement such that, by sensing an entirety of a first of the data structures and at least a predetermined portion of at least one other data structure near the first data structure, the entire distributed codeword can be reconstructed.

13.   A plurality of coded data structures according to claim 12, each of the coded data structures conforming to one of two or more formats, wherein a first of the formats includes a first portion of the distributed codeword, and each of the second and any subsequent formats includes respective second and subsequent portions of the distributed codeword, at least one coded data structure of each of the formats being disposed on or in the substrate such that at least the first coded data structure and a predetermined portion of at least one of each of the second and any subsequent coded data structures can be sensed to enable the distributed codeword to be reconstructed.

14.   A plurality of coded data structures according to claim 12 or 13, wherein each of the coded data structures encodes information indicative of a rotational orientation of that coded data structure.

15.   A plurality of coded data structures according to any one of claims 12 to 14, wherein the distributed codeword is indicative of an identity.

16.   A plurality of coded data structures according to claim 5, wherein the identity is an identity of an electronic document.

17.   A method of sensing coded data structures disposed on or in a substrate to determine at least a distributed codeword, wherein each of the coded data structures includes at least a local codeword and some portion of the distributed codeword, the method including the steps of:
(a)   sensing an entirety of a first of the coded data structures and at least a part of at least one other data structure near the first data structure;
(b)   reconstructing, from the at least one sensed part and the codeword portions therein, the distributed codeword.

18.   A method according to claim 17, wherein each of the coded data structures conforms to one of n formats where n is at least 2, each of the formats including a portion of the distributed codeword, wherein:
        step (a) includes sensing a part of at least one coded data structure of each format; and
        step (b) includes reconstructing, from the sensed parts and the codeword portions therein, the distributed codeword.

19.   A method according to claim 18, wherein each of the coded data structures encodes information indicative of a rotational orientation of that coded data structure, the method including the steps, after step (a), of:
        determining a rotational orientation of the first coded data structure;
        identifying the format of the first coded data structure;
        identifying the format of each of the other coded data structures to which the sensed parts correspond.

20.   A method according to any one of claims 17 to 19 wherein the distributed codeword is indicative of an identity.

21.   A method according to claim 20, wherein the identity identifies an electronic document.

22.   A substrate bearing coded data structures according to any one of claims 12 to 16.

23.   A substrate according to claim 22, wherein the coded data structures are printed onto a surface of the substrate.

24.   A substrate according to claim 22 or 23, wherein the coded data structures are printed in an infra-red absorbent ink that is invisible or substantially invisible to an average unaided human.

25.   A method of manufacturing a substrate according to any one of claims 22 to 24, including printing one or more of the coded data structures onto a surface of the substrate.